THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today
(1) was not written for publication in a law journal and
(2) is not binding precedent of the Board.

Paper No. 21

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte HIDEKI KOYANAGI and
TOHRU WADA

Appeal No. 96-2696
Application 08/167,415[1]

ON BRIEF

Before MARTIN, JERRY SMITH, and RICHARD TORCZON, *Administrative Patent Judges*.

MARTIN, *Administrative Patent Judge*.

DECISION ON APPEAL

[1] Application for patent filed December 14, 1993.

This is an appeal under 35 U.S.C. § 134 from the examiner's rejection of claims 1-15, 17-19, 21, 23, and 24, all of the pending claims, over prior art.[2] We reverse.

The invention relates to encoding and decoding for use in transmitting and reproducing compressed motion image data (Spec. at 1, lines 5-9).

The background of the invention

Appellants' Figures 1(A)-1(C) show a prior art image encoding apparatus (Spec. at 22, lines 3-5). This system includes, inter alia, a motion vector detecting circuit 1 responsive to image information stored in "forward original image part" 2a, "reference original image part" 2b, and "backward original image part" 2c, an arithmetic operation part 3 (Spec. at 3, lines 15-16), a DCT (discrete cosine transformation) circuit 4 (Spec. at 5, lines 6-7), a quantizing circuit 5, a variable length encoding circuit 6, a send buffer 7, and a transmitting data control circuit 111.

[2] Page 5 of the Answer incorrectly includes claim 22 among the rejected claims. Claim 22 was canceled by an amendment after final received April 11, 1995 (paper No. 9).

The quantizing circuit is responsive to the amount of data in the send buffer (Spec. at 5, lines 7-12).

As shown in appellants' Figures 2(a) to 2(c), each frame of image data is divided into N slices, each of which contains M macroblocks (also referred to as macro blocks), each of which in turn consists of luminance data Y1 to Y4 and color-difference signal data Cb and Cr (Spec. at 3, lines 13-23). Appellants' specification explains that prior art systems have inserted "invalid codes" into the data stream at the transmitting end in order to prevent an underflow of data:

> [S]ince the image data is transmitted as the variable length code, when a simple stationary image, for example, continues for a comparatively long[] period, the data to be transmitted becomes shortage [sic]. In this case, in view of preventing missing . . . transmission data, an invalid code can be added to the data to be transmitted. This invalid code can also be added, for example, in units of [a] slice or macroblock shown in Figs. 2(a) to 2(c). [Spec. at 13, lines 4-12.]

Examples of such invalid codes are described as follows with reference to appellants' Figures 3 and 4:

> Fig. 3 illustrates an example where an invalid code (invalid data) is added in units of [a] slice. Each slice is provided with a slice start code at its leading area. This slice code is formed by a synchronous code and an attribute code. The synchronous code is formed by the data of two bytes where each bit is all [sic] set to logic 0 and the data of one byte (three bytes in total)

> where LSB is set to logic 1 and the other bits are set to logic 0. Moreover, the attribute code is set to one byte where the code indicating the data concerning the slice data such as the attribute of [the] corresponding slice is arranged. Therefore, the slice start code is formed by the data of 4 bytes in total (32 bits).
>
> This slice start code is added in a unit of [a] slice without relation to [a] shortage of the data to be transmitted.
>
> The invalid code is added in such a manner that the data where all the bits are set to logic 0 is added before the slice start code as required in unit[s] of the byte (8 bits). This code is added in units of [a] slice only when there is a shortage of data to be transmitted.
>
> Fig. 4 illustrates an invalid code to be added to a c block [identified in the figure as a macro block]. Namely, in this case, [a] total of 11 bits where [the] upper 7 bits are set to logic 0 and [the] lower 4 bits to logic 1 are considered as a unit of the invalid code and this invalid code is added before a valid code of the macroblock as many as the predetermined number of units. [Spec. at 13-14.]

Where invalid code is added to the data of a macroblock in the manner shown in Fig. 4, the transmission data control circuit 111 of Figure 1 can take the form shown in Figure 5 (Spec. at 14, lines 19-22). In figure 5, MUX (multiplexer) 122, which is controlled by a controller 124, receives the output of macro block invalid code generating circuit 123 and the output of N/M converter 121, which receives data from transmitting or send buffer 7 (Spec. at 14, line 22 to p. 15, line 6). Thus, the multiplexer combines data from the N/M converter with invalid code generated by macro block invalid code generating circuit 123 (Spec. at 15, lines 2-6). The controller 124

> controls the multiplexer 122 corresponding to the send buffer information and selects, when the send buffer 7 does not generate [an] underflow condition, an output of the N/M converter 121 or selects an invalid code outputted from the macroblock invalid code generating circuit 123 when the send buffer 7 is supposed to generate [an] underflow condition. Therefore, the data outputted from the multiplexer 122 mixes invalid codes of the desired number of units. [Spec. at 15, lines 12-20).

Thus, an underflow condition sensed in send buffer 7 is corrected by inserting invalid codes into the output data stream provided by the send buffer.

Figures 6(A) and 6(B) show a prior art decoding apparatus for decoding the data generated by the encoding apparatus of Figures 1(A)-(C) (Spec. at 15, lines 21-24). The received encoded data is temporarily stored in receiving buffer 32 and then supplied to a variable length decoding circuit 33, the output of which is processed by inverse quantizing circuit 34 and then IDCT circuit 35 (Spec. at 15, last line to p. 16, line 22). The variable length decoding circuit 33 removes any invalid codes inserted by the encoding apparatus at the transmitting end (Spec. at 19, lines 15-21). However, the following discontinuity problem results:

> [W]hile the invalid code is being removed in the variable length decoding circuit 33, data is not supplied to each circuit after the inverse quantizing circuit 34 in the successive stages, bringing about a disadvantage that these circuits are not used for the processing. In the case of the NTSC system, an image of one frame is displayed on the display[] 40 in the period of 1/30 second. But, if the invalid code is long, each circuit after the inverse quantizing circuit 34 cannot process the data of one frame within the period of 1/30 second and display of [an] image on the display 40 has been discontinued. [Spec. at 19, line 22 to p. 20, line 9.]

We understand this passage to mean that the time it takes for the variable length decoder to eliminate a long series of invalid codes from the transmitted data stream causes a discontinuity in the displayed image.

Appellants' invention

Appellants avoid the foregoing discontinuity problem by removing invalid codes from the received encoded image data before it reaches receiving buffer 32 and variable length encoding circuit 33 (Spec. at 20, line 11 to p. 21, line 9). This is accomplished by using the invalid code format shown in Figure 9 for a macroblock (Spec. at sentence bridging pp. 25 and 26). This code format, which is referred to in the specification at page 27, lines 8-12 as "invalid code in the broad meaning," consists of a staffing start code and adjusting data (Spec. at 26, lines 1-3). The staffing start code "is formed of a synchronous code and a staffing start [sic, staffing] code" (Spec. at 26, lines 3-4). The synchronous code is "a unique pattern and it is determined so that any other pattern [the] same as the synchronous code is never generated" (Spec. at 26, lines 8-11). The staffing code includes information about the data included in the macroblock (Spec. at 26, lines 11-15). The adjusting data consists of a synchronizing code byte followed by the "required number" of all-zero bytes representing invalid codes (in the narrow sense) (Spec. at 26, line 16 to p. 27, line 7), i.e., the number of invalid codes needed to avoid the above-described underflow condition.

Figures 7(A) to 7(C), which show appellants' encoding apparatus for inserting the invalid codes shown in Figure 9, is the same as the prior art encoding apparatus shown in Figures 1(A) to 1(C), except that appellants' transmitting data control circuit 8 (Fig. 7(C)) differs from the prior art transmitting data control circuit 111 of Figure 1(C) (Spec. at

24, line 22 to p. 25, line 9). Appellant's transmitting data control circuit 8, which generates the invalid codes of Figure 9, is shown in Figure 8 and includes a invalid code generating circuit 23 for generating the adjusting data and a staffing start code generating circuit 24 for generating the staffing start code (Spec. at 27, lines 16-21). The Figure 8 circuitry is similar to its prior art counterpart (Figure 5) in that it adds invalid codes to the output of send buffer 7 of Fig. 7(C) when an overflow condition is detected in the send buffer: "[I]f the data of a macroblock is likely to generate underflow corresponding to the information sent from the send buffer 7, the multiplexer 22 is controlled and the invalid code (of broad meaning) is outputted to the transmission path" (Spec. at 29, line 23 to p. 30, line 3).

Figures 11(A) and (B) show appellants' decoding apparatus. In contrast to the prior art decoding apparatus shown in Figures 6(A) and (B), appellant's decoding apparatus includes an invalid code eliminating circuit 31 for eliminating invalid codes before they reach the receiving buffer 32 (Spec. at 30, lines 4-16). The details of invalid code eliminating circuit 31 are shown in Figure 12 (Spec. at

30, lines 17-18) and its operation is described with the aid of Figures 13(a)-(e). Figure 13(b) shows a received encoded bit stream in which each byte is represented as a hexagon (Spec. at 31, lines 21-24), identified either by the last two digits of its code (00 or 01) or by the letters "stf," apparently representing the staffing code. In Figure 13(c), which shows these bytes after conversion in M/8 converter 51 and a one-clock cycle delay in 8-bit shift register 52, the received bytes are labeled C_1-C_9. Comparing these bytes to the format of Figure 9, byte C_1 (00) corresponds to the synchronization code part of the adjusting data, bytes C_2-C_5 (all 00) correspond to the invalid codes of the adjusting data, bytes C_6-C_8 (00, 00, 01) correspond to the synchronous code part of the staffing start code, and byte C_9 (stf) corresponds to the staffing code part of the staffing start code. The circuitry of Figure 12 allows only the first three 00 bytes (C_1-C_3) of the series of seven 00 bytes (C_1-C_7) to be written from the 8/L converter 53 to the receiving buffer 32 (not shown) by inhibiting further writing until detection of the first non-zero byte, which is an 01 byte (C_8), at which time writing resumes, thereby allowing the writing of 01 byte

C_8 and stf byte C_9 into the receiving buffer (Spec. at 31, line 21 to p. 33, line 13). Thus, the Figure 12 circuitry eliminated four 00 bytes, i.e., bytes C_4-C_7, which is equal to the number of bytes that represented adjusting data in the received encoded data. In so doing, the circuitry also changed the identity of the bytes which represent the four-byte staffing start code, which was initially represented by bytes C_6-C_9 in the received encoded data stream but then by bytes C_2, C_3, C_8, and C_9 in the modified data stream, which is written into the receiving buffer 32 (Spec. at 33, lines 13-16; p. 34, lines 12-14).

The result of eliminating four of the 00 bytes prior to storage of the data in the receiving buffer is that the variable length decoder 33 (Fig. 11(A)) is required to eliminate only three 00 bytes, i.e., synchronizing byte C_1 and bytes C_2 and C_3 of the staffing start code (Spec. at 34, lines 12-21). This permits the encoded image data to be processed effectively (Spec. at 33, line 22 to p. 35, line 2), i.e., without causing a discontinuity in the displayed image.

The claims

Claim 13, which is representative of the claims that are directed to the encoding end, reads as follows:

> 13. An image encoding method, comprising the steps of:
>
> dividing image data of one frame into a plurality of macroblocks, one macroblock consisting of a two-dimensional arrangement of a plurality of pixels;
>
> encoding data of said macroblock by using a predetermined data compressing method; and
>
> adding an invalid code for preventing underflow of data in a send buffer for the compressed data of said macroblock,
>
> wherein said invalid code is formed with a staffing start code consisting of synchronous codes specified to not allow generation of the same pattern in the image data and a staffing code indicating information about an attribute of said macroblock.

Claim 1, which is representative of the claims that are directed to the decoding end, reads as follows:

> 1. An image decoding apparatus for decoding transmitted compressed image data, comprising:
>
> a receiving buffer for temporarily storing said compressed image data;
>
> decoding means for reading compressed image data stored in said receiving buffer corresponding to proceeding [sic, preceding] conditions of a decoding process and decoding image data of one frame within a period no longer than a frame period of the image; and

eliminating means coupled to said receiving buffer to supply said compressed image data to said receiving buffer after eliminating invalid code for preventing underflow of data in a send buffer provided in an image coding apparatus.

Both of these claims recite "invalid code for preventing underflow of data in a send buffer." As explained above, in both the prior art system and appellants' invention an underflow condition is detected in the send buffer and corrected by inserting invalid code into the output data stream of the send buffer.

The references

The examiner relies on the following references (Answer at 2):

Sun et al. (Sun)	5,247,363	Sep. 21, 1993 (filed Mar. 2, 1992)
Acampora et al. (Acampora)	5,287,178	Feb. 15, 1994 (filed Jul. 6, 1992)

The rejections

Claims 1, 2, 4-7, 9-10, 12 and 24 stand rejected under 35 U.S.C. § 102(e) as anticipated by Sun.

Claims 3, 8, and 11 stand rejected under § 103 as unpatentable for obviousness over Sun.

Claims 13-15, 17-19, 21, and 23 stand rejected under § 102(e) as anticipated by Acampora.

The rejections based on Sun

Sun, which discloses an error concealment method for HDTV receivers, explains in the "Background of the Invention" section:

> The Advanced Television Research Consortium (ATRC) in the United States has developed an HDTV system based upon the MPEG format, for transmission of high definition television (HDTV) signals in digital form. This HDTV System is described in U.S. Pat. No. 5,122,875. In the HDTV system developed by the ATRC, video signal compressed according to an MPEG like format is arranged in service type specific transport packets for transmission. These packets undergo a first level of error encoding, to generate a frame check sequence FCS, e.g., a cyclic redundancy check, and FCS error check codes are appended to the transport packets. Thereafter the transport packets, with the appended error check codes, undergo a forward error coding, FEC, such as a Reed-Solomon coding, and FEC error detection/correction codes are appended to the data.
>
> At the receiver, transmitted information is detected and applied to an FEC decoder, which performs a limited error correction function on the transmitted data. The FEC corrects the majority of errors incurred during transmission[;] however because the error check overhead volume is limited by bandwidth constraints, some errors will pass the FEC decoder without detection/correction. The FEC decoded signal is then coupled to an FCS decoder. The FCS decoder is capable of detecting but not correcting data. If an error is detected in a transport packet, the entire transport packet is discarded.
>
> The discarded transport packets may thereafter be replaced with synthesized compressed data. An example of apparatus for performing pre-decompression error concealment is described in copending application Ser.

> No. 07/789,245, filed Nov. 7, 1991 and entitled "Apparatus For Concealing Errors In A Digital Video Processing System." In this system the FCS decoder generates error tokens which indicate when data has been discarded. Responsive to the error tokens the system substitutes predetermined sequences of compressed data for discarded data. The data is then decompressed for display or storage purposes. Depending upon the type of data that is lost, replacement by synthesized compressed data may not produce satisfactory images. [Col. 1, line 31 to col. 2, line 5.]

In accordance with Sun's invention,

> [w]hen an image area is identified with data that has been discarded, adjacent decompressed image areas are examined for motion and detail. Depending upon the relative amount of motion or detail in adjacent image areas, the current image area is replaced with spatially synthesized or temporally co-located data respectively. [Col. 2, lines 13-19.]

The examiner contends that the term "invalid code" as used in appellants' claims is a very broad term and that

> [a]ny non-information data appended to the information data packet can be considered as "invalid code". For instance, forward error correction code (FEC), frame check sequence code (FCS), and synchronization code, which are all packed as part of a header code or transport packet, to perform error correction, redundancy check and synchronization respectively, are effectively removed at the receiving side during decompression (see col. 4, lines 67-68). [Answer at 7.]

We agree with appellants that the examiner's broad construction of "invalid code" ignores the fact that the claims specify that the invalid code is used to prevent the underflow of data in a send buffer, which is not the purpose of any of the codes cited by the examiner.[3] Consequently, we agree with appellants that Sun also fails to disclose or suggest the claimed eliminating means coupled to a receiving buffer to supply compressed image data to the receiving buffer after eliminating such invalid code.

We note that the examiner (Answer at 7), in addressing appellants' argument that the claim calls for eliminating the invalid code <u>prior to decoding</u> of the compressed image data, characterized this argument as inconsistent with the fact that the claim recites <u>decoding</u> apparatus <u>comprising</u> an <u>eliminating</u> means for eliminating invalid code. We do not agree. The fact that the preamble recites decoding apparatus does not preclude the claimed eliminating means from removing invalid code from the compressed image data before it is decoded by the claimed decoding means, which receives the compressed image data from the claimed receiving buffer.

[3] While such codes are used in the admitted prior art described in appellants' application, that prior art is not relied on in any of the rejections before us.

For the foregoing reasons, the § 102(e) rejection of claim 1 based on Sun is reversed, as are the remaining rejections based on Sun, i.e., the § 102(e) rejection of independent claims 9 and 24, which recite limitations similar to those recited in claim 1, the § 102(e) rejection of dependent claims 2, 4-7, 10, and 12, and the § 103 rejection of dependent claims 3, 8, and 11.

The rejection based on Acampora

Claim 13, which is rejected as anticipated by Acampora, is directed to the encoding method used in appellants' disclosed invention. This claim recites the steps of, inter alia, encoding data of a macroblock by using a predetermined data compressing method and

> adding an invalid code for preventing underflow of data in a send buffer for the compressed data of said macroblock, wherein said invalid code is formed with a staffing start code consisting of synchronous codes specified to not allow generation of the same pattern in the image data and a staffing code indicating information about an attribute of said macroblock.

Acampora discloses an MPEG-like HDTV encoder (col. 2, lines 36-40) which employs a reset control means for ensuring that a header is properly merged with a valid data word rather than with extraneous data, or, conversely, that data words are properly merged with a valid associated header rather than with extraneous header bits (col. 1, lines 58-63). As with Sun, the examiner (Answer at 9) contends that the term "invalid code" is broad enough to read on any non-information data or portion of the header data and thus reads on the GOP (group of pictures) header data. This ignores claim 13's recitation that the invalid data is used to "prevent[] underflow of data in a send buffer in an image coding apparatus," which is not true of Acampora's GOP headers. Acampora fails to disclose or suggest using such invalid codes, let alone forming them with the specific format recited in the claim.

The § 102(e) rejection of claim 13 over Acampora is therefore reversed, as are the other § 102(e) rejections based on that reference, i.e., the rejection of independent claims 17 and 21, which recite limitations similar to those of claim 13, and the rejection of dependent claims 14, 15, 18, 19, and 23.

REVERSED

)
JOHN C. MARTIN)
Administrative Patent Judge)
)
)
) BOARD OF PATENT
JERRY SMITH)
Administrative Patent Judge) APPEALS AND
)
) INTERFERENCES
)
RICHARD TORCZON)
Administrative Patent Judge)

cc:

Philip M. Shaw, Jr., Esq.
Limbach & Limbach
2001 Ferry Building
San Francisco, CA 94111-4262

dm